

1. JP,2923002,B

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

(57) [Claim(s)]

[Claim 1] In the raster image processing system which processes the data described by the Page Description Language, and forms a raster image this system A raster image processing means to receive said data, to create the font data assignment information for obtaining the font data of the alphabetic character contained in these data, and to form a raster image from the obtained font data, The profile data showing a character font access the database accumulated in each format which can be used [equipment] based on font data assignment information. The font management tool which incorporates the profile data relevant to this font data assignment information, The interface means which carries out the interface of between said raster image processing means and font management tools in the common format independent of a font is included. Said font management tool It has the 1st are recording means which stores said profile data temporarily. This font management tool If the 1st are recording means is accessed first and the profile data relevant to this font data assignment information exist in the 1st are recording means according to said font data assignment information When font data is generated based on this and the profile data relevant to this font data assignment information do not exist in the 1st are recording means While accessing said database based on this font data assignment information, being based on the profile data relevant to this font data assignment information being incorporated and generating font data the font characterized by storing these profile data that were crowded picking in the 1st are recording means -- a free raster image processing system.

[Claim 2] In a system according to claim 1 said font management tool It has the 2nd are recording means which stores temporarily the rasterized alphabetic data which is contained in said formed raster image. In case said raster image processing means forms a raster image, the rasterized alphabetic data is accumulated in the 2nd are recording means. If the 2nd are recording means is accessed in advance of access to the 1st are recording means and the rasterized alphabetic data relevant to this font data assignment information exists in the 2nd are recording means according to said font data assignment information It is the raster image processing system characterized by accessing the 1st are recording means based on this font data assignment information when font data is generated based on this and the rasterized alphabetic data relevant to this font data assignment information does not exist in the 2nd are recording means.

[Claim 3] It is the raster image processing system characterized by including the 1st storage region where the 1st are recording means was beforehand defined among said profile data in the system according to claim 1, and which accumulates relatively what has high operating frequency in resident, and the 2nd storage region which accumulates relatively what has low operating frequency in transient.

[Claim 4] It is the raster image processing system characterized by including a database means to output the profile data relevant to it if it is accumulated in the common format that said profile data do not depend for this system on a font in a system according to claim 1 and said font data assignment information is received.

[Claim 5] It is the raster image processing system characterized by including the host who creates the data with which this system was described by said Page Description Language in the system according to claim 1, and supplies said raster image processing means.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

(Field of the invention on industry)

This invention relates to a raster image processing system and the raster image processing system which more specifically processes the data described by the Page Description Language, and forms a raster image.

(Prior art)

The command data containing an alphabetic character and image data are created for example, by the application program for publishing, and are changed into the medium stream described by the Page Description Language. This medium stream is changed into a raster image by the interpreter by the raster image processing (RIP) method, and is outputted to image output units, such as a printer and an indicating equipment, in the form of a page.

In the system handling an alphabetic character, various fonts exist according to various language and countries, therefore there are various demands to a font. For example, there is versatility, such as the alphabetic character, a typeface, and not only magnitude but lateral writing, columnar writing, left writing, right writing, a slanting font, a bold letter, etc. Moreover, when treating a font as a character string, i.e., a document, array methods, such as the method of the array of a character string, for example, proportional-spacing printing etc., also have various demands. Furthermore, various character code sequences are used by the feeder of a font about the code which specifies an alphabetic character.

Raster image mode of processing must carry out satisfaction of these various requirements selectively at least. In addition, in this description, the vocabulary "a font" shall be interpreted in a wide sense not only including the alphabetic character or a font but above various requirements.

A font feeder is provided with alphabetic-character data in a format, for example, the bit map format, an outline (profile) format, etc. of a proper. Generally font data is accumulated in memory, such as a foreign file and ROM, in the form of the digitized data where the straight line and curve of a profile of an alphabetic character are specified. If a font is specified by the host by the specific character code, reading appearance of the font data corresponding to it will be carried out from a file, and scaling processing of amplification or a cutback will be performed to it, it will be rasterized eventually, and will be changed into an image.

The processing which develops the original data based on a bit map format or an outline format to a raster image is different with the format of a font in many cases. Moreover, after carrying out the scaling of the original data, the HINTINGU processing which amends the length and a radius of circle and is made into a suitable typeface may also change with font formats.

Conventionally, raster image mode of processing of a proper, i.e., mode of processing of a font orientation mold, was taken by the classification of a font under the environment of such publishing. The conventional printer is connected more to a detail with general interfaces, such as Centronics and RS-232C, at hosts, such as a personal computer and a workstation, including a raster image processor and a font data file. That is, the raster image processor and font data in a printer are peculiar to a specific printer, and the configuration for which a host uses them through a general-purpose printer interface was taken. Moreover, although image display devices, such as CRT, were connected to the host, the configuration for which the data file of the font used with an indicating equipment and the raster image processor treating it are contained inside a host, therefore it depends on a host was taken.

(Technical problem which invention tends to solve)

Conventionally, the single raster image processing system which can respond to various requirements mentioned above accommodative was not able to be used so that it may understand from now on. The raster image processor of a proper was required for it to the command data which specific application software uses for a detail more. For example, a PostScript interpreter is used for the command data stream by Page Description Languages, such as PostScript

(trademark), as a raster image processor. That is, a font format, specific font processing, and a specific font read-out function were included in the specific raster image processor.

Generally, in document handling systems, such as publishing, two or more command data streams are used. Therefore, in the conventional method, in order to make possible treatment of two or more data streams, the font had to be prepared for every command stream. This caused processing of a font, and duplication creation of a managerial system, and was carrying out induction of the duplication capitalization of a font.

By the way, in order to accelerate the display of an alphabetic character, when the alphabetic data which was used once and rasterized was stored in a cache and the display output of the same alphabetic data was carried out after that conventionally, there was a system which reads and uses this from a cache. By this, the alphabetic data by which a multiple-times activity is carried out does not need to perform processing which rasterizes font data.

However, as for the rasterization alphabetic data accumulated in a cache, size is fixed to that condition of having been used, by this conventional method. Therefore, when using the same character font as a degree in different size, the alphabetic data accumulated in the cache could not be used, but alphabetic character font data had to be separately fetched from the file. Probably, many storage regions are needed, though natural, supposing it constitutes a system so that the alphabetic data rasterized in two or more sizes about the same character font may be temporarily accumulated in a cache. This inclination is remarkable about a font with many types of letters like the kanji.

This invention can cancel the fault of such a conventional technique, high-speed character representation can be performed, and it aims at offering the raster image processing system independent of a font.

(The means for solving a technical problem)

According to this invention, the raster image processing system which processes the data described by the Page Description Language, and forms a raster image A raster image processing means to receive said data, to create the font data assignment information for obtaining the font data of the alphabetic character contained in this data, and to form a raster image from the obtained font data, The profile data showing a character font access the database accumulated in each format which can be used [equipment] based on font data assignment information. The font management tool which incorporates the profile data relevant to font data assignment information, The interface means which carries out the interface of between a raster image processing means and font management tools in the common format independent of a font is included. A font management tool It has the 1st are recording means which stores profile data temporarily. A font management tool If the 1st are recording means is accessed first and the profile data relevant to font data assignment information exist in the 1st are recording means according to font data assignment information When font data is generated based on this and the profile data relevant to font data assignment information do not exist in the 1st are recording means A database is accessed based on font data assignment information, and while being based on the profile data relevant to font data assignment information being incorporated and generating font data, the incorporated profile data are stored in the 1st are recording means.

Moreover, according to this invention, a font management tool has the 2nd are recording means which stores temporarily the rasterized alphabetic data which is contained in the formed raster image. In case a raster image processing means forms a raster image, the rasterized alphabetic data is accumulated in the 2nd are recording means. If the 2nd are recording means is accessed in advance of access to the 1st are recording means and the rasterized alphabetic data relevant to font data assignment information exists in the 2nd are recording means according to font data assignment information When font data is generated based on this and the rasterized alphabetic data relevant to font data assignment information does not exist in the 2nd are recording means, the 1st are recording means is accessed based on font data assignment information.

Furthermore, according to this invention, the 1st are recording means includes the 1st storage region which was beforehand appointed among profile data and which accumulates relatively what has high operating frequency in resident, and the 2nd storage region which accumulates relatively what has low operating frequency in transient.

(Operation)

If the data described by the Page Description Language are inputted into a raster image processing system from a host according to this invention, a raster image processing means will create the font data assignment information for obtaining the font data of the alphabetic character contained in this data. A font management tool accesses the 1st are recording means first according to font data assignment information. If this profile data exists in the 1st are recording means, font data will be generated based on this. However, when it does not exist, a database is accessed based on font data assignment information, the profile data relevant to the font data assignment information are incorporated, and font data is generated based on this. The profile data incorporated with this are stored in the 1st are recording means.

A raster image processing means forms a raster image from the profile data obtained in this way. The formed raster image is eventually outputted to a raster image output unit or a communication device.

(Example)

Next, with reference to an accompanying drawing, the example of the raster image processing system by this invention is explained to a detail. As for this example, fundamentally, reference of drawing 1 connects both mutually by connectors 14 and 16 including the module package 12, i.e., the RIP processing module, which performs raster image processing (RIP) with the host 10 who consists of processing systems, such as a personal computer and a workstation. A host 10 is a processor general-purpose in this example which creates the command data containing an alphabetic character and image data with the application program 50 (drawing 2) for publishing, and changes them into the medium stream described by the Page Description Language.

The image display device 18 which indicates an alphabetic character and images, such as CRT, by visible is connected to the host 10. moreover, the printer 20 is connected to the host 10 and (or) the RIP processing module 12. A printer 20 is an image output unit which carries out visible record of an alphabetic character and the image at a record medium. Of course, although a host 10 also contains I/O devices, such as a keyboard, like the usual processing system, since these are not directly related to an understanding of this invention, they omit a graphic display and explanation.

It is characteristic by the indicating equipment 18 and the printer 20 that the font data used common to visualization of an alphabetic character is stored in the foreign files 22, such as a fixed disk, a floppy, or an integrated-circuit board, in this system. That is, it is not dependent on an indicating equipment 18 or a printer 20, and these font data can be used in common as a database. Of course, what can access a host 10 not only through the store of a proper but through a communication line at a distant database is sufficient as a foreign file 22. Therefore, while a font is not dependent on a RIP processor, for example, there is wide range expandability from roman to the kanji, on the other hand, a RIP processor becomes independent of a character coding scheme.

The RIP processing module 12 contains a RIP processing system so that it may explain in full detail behind. This point is also one of the important descriptions of this example. A host 10 can perform various application software 50 for publishing. Therefore, according to those application software 50, the command data stream 24 of various classes is inputted into the RIP processing module 12. Moreover, reading appearance of the font data of a foreign file 22 is carried out dynamically if needed for the RIP processing module 12, and it is loaded to the RIP processing module 12. The RIP processing module 12 performs suitable raster image processing according to the class to the command data stream 24, and outputs it to a host 10 or a printer 20 as raster image data 26. Even if this data 26 is the form by which the data compression was carried out, it may be a form which is not compressed.

The RIP processing module 12 includes the specific configuration for two function parts, the RIP processing 52 and the font management 54, fundamentally, as shown in drawing 2 . The interface of the application software 50 is carried out to the RIP processing 52 with the software interface 56.

The RIP processing 52 creates font data assignment information required to take out the font data of the alphabetic character specified with specific application software 50. By this example, the typeface and character coding scheme based on assignment of a font feeder, a character code, a character size, etc. are contained in this font data assignment information. With a typeface, if it is a kanji, for example, if it is Japanese, by the character coding scheme, the exception of JIS83 or Shift JIS etc. will be specified exceptions, such as a Mincho typeface and Gothic.

The interface 58 between the RIP processing 52 and the font management 54 takes the communalized interface independent of a font. An interface 58 contains above-mentioned font data assignment information and font data. Font data is loaded to the field 60 in a module 12 from the database of a foreign file 22 according to a demand. Therefore, the font access path in the font management 54 can take the optimal pass according to a character coding scheme. The font management 54 performs font processing of expanding or reducing font data 60 to the size which is the form of font data as it is, or is demanded by character-size data, and outputs font data to the RIP processing 52.

In this example, the storage region 60 of font data includes two high-speed storage regions, the profile alphabetic character cache 100 and the rasterization alphabetic character cache 102, as shown in drawing 3 . The profile alphabetic character cache 100 is a field where it is stored temporarily, the data, i.e., the profile data, showing the profile of the character font read from the font database 22. This profile data is a size free-lancer fundamentally, and it is used to the alphabetic data with which desired magnitude was rasterized for behind, changing it.

The rasterization alphabetic character cache 102 is a storage region where the alphabetic data of a result with which the profile data stored in the profile alphabetic character cache 100 were rasterized by the image of desired magnitude is accumulated. Variable power of the image data stored in the rasterization alphabetic character cache 102 is carried out to desired magnitude. In addition, rasterization is performed by the RIP processing 52 like the after-mentioned. Thus, in this example, the profile data read from the font database 22 are stored in the profile alphabetic character cache 100, and the alphabetic character cache is hierarchized by two level so that the alphabetic data rasterized next may be stored in the rasterization alphabetic character cache 102. However, the rasterization alphabetic character cache 102 is very good

in the system configuration which does not necessarily need to be prepared and contains only the profile alphabetic character cache 100 about an alphabetic character cache.

Further, the profile alphabetic character cache 100 includes two fields 104, i.e., a resident area, and a transient area 106 by this example, as shown in drawing 4. A resident area 104 is a storage region which stores the profile data of a character font with high operating frequency relatively like a hiragana for example, in a Japanese system. The profile data of an alphabetic character with high operating frequency are read into a resident area 104 from the font database 22 in the case of initial setting of a system. The profile data read into the resident area 104 are always held and used for this field 104.

A transient area 106 is a storage region which stores the profile data of a character font with low operating frequency relatively. For example, the alphabetic data of the kanji is accumulated in this field 106, when the specific kanji is accessed by the font database 22 for the first time. The storage capacity of a transient area 106 is limited. Then, there is a method of being the FIFO actuation which eliminates the alphabetic data read in ancient times, when it accumulates sequentially from what was most read into recently, for example and is accumulated to the limit of memory capacity, and performing read-and-write control per alphabetic character. Or counting of each use count of the profile data stored in the transient area 106 may be carried out, and you may constitute so that it may eliminate sequentially from the fewest data of a use count.

As for the font management 54, in this example, being based on management of an one-character unit is desirable. This is for making the access mechanism of the font data which it had by the command stream according to each application software with the conventional technique absorb by the font management 54 in this example. When the scaling of the font data is carried out, the die length of the segment which forms the profile of an alphabetic character may become irregular, or smoothness may lack in a radius of circle. In such a case, HINTINGU processing which amends die length and a radius of circle and is made into a suitable typeface is also performed by the font management 54.

The RIP processing 52 rasterizes carrier beam font data from the font management 54. The rasterized image data is accumulated in an image / frame memory 62, and is assembled by the image data of a page unit. An image / frame memory 62 is storage regions which store raster image data for example, per page temporarily.

The interface of an image / the frame memory 62 is carried out to a printer 20 and a display 18 through the video interface 64. Moreover, an interface is carried out by picture compression 68 to output units by which an interface is carried out in a compressed data format, such as printer 20a and facsimile 66. To a host 10, the RIP processing module 12 is connected by bus, and this bus connection is performed through the hardware bus interface 70.

Actuation is explained. First, if a power source is supplied to equipment, the RIP processing 52 will access the font database 22, will read relatively the character font of high operating frequency, for example, the profile alphabetic data of a kana, into the profile alphabetic character cache 100, will accumulate these in the resident area 104, and will equip a next activity with them.

The command data containing an alphabetic character and image data are created by application software 50, and are changed into the medium stream described by the Page Description Language. This medium stream is inputted into the RIP processing 52 through the software interface 56. In the RIP processing 52, font data assignment information required to access the font data of the alphabetic character contained in this command stream is created, and this is given to the font management 54.

The font management 54 accesses the rasterization alphabetic character cache 102 first based on font data assignment information. If the rasterization alphabetic data of a required character size exists in a cache 102, the font management 54 will incorporate it. If there is no rasterization alphabetic data of a required character size in a cache 102, the font management 54 will access the profile alphabetic character cache 100, and it will be incorporated if the data exists in a cache 100. For example, a hiragana always comes to hand in the form of profile data from this cache 102 at this example. When required profile data exist in neither of both the caches 100 and 102, the font management 54 is accumulated also in the profile alphabetic character cache 100, and next utilization is equipped with it while it accesses the font database of a foreign file 22 and incorporates required font data. For example, the profile data is read into the transient area 106 of the profile alphabetic character cache 100 about character fonts with comparatively low operating frequency, such as a kanji.

It is the form of font data as it is, or expands or reduces to required size, and the font management 54 outputs the font data of the profile alphabetic character cache 100 to the RIP processing 52. Reading appearance of the other profile data is carried out for the profile data with high operating frequency, such as a kana, from a transient area 106 from a resident area 104.

In the RIP processing 52, carrier beam font data is rasterized from the font management 54, and it once accumulates in an image / frame memory 62 as document data. In that case, the rasterized character font is stored also in the

rasterization alphabetic character cache 102 of the font data storage region 60, and subsequent activities are equipped with it. In this way, the image data of a page unit is assembled by an image / frame memory 62.

Coding compression is carried out by picture compression 68 when the raster image accumulated in the image / frame memory 62 is outputted through the video interface 64 when outputting to a printer 20 or an indicating equipment 18, and outputting to communication devices, such as printer 20a or facsimile 66. In this way, the command data stream from application software 50 is changed into a raster image by the RIP processing module 12, for example, is outputted to image output units, such as a printer 20 and an indicating equipment 18, in the form of a page.

At this example, in this way, the profile alphabetic character cache 100 is formed and font data is accumulated in this in the form of profile alphabetic data before being rasterized. This profile alphabetic data is accessed at every need for an activity behind, variable power of it is carried out to desired magnitude, it is rasterized, and is outputted as an image data. Therefore, as compared with the method with which only the conventional rasterization cache was formed, it can be adapted for all character sizes with small alphabetic character cache storage capacity. In this example, since the profile alphabetic character cache 100 has taken the layered structure with the rasterization alphabetic character cache 102, an efficient alphabetic character cache is realized again.

(Effect of the invention)

According to this invention, the cache which stores the profile data of a character font temporarily is formed in this way, and font data is accumulated in this in the form before being rasterized. The profile data stored in the cache are accessed at every need for an activity behind, variable power of them is carried out to desired magnitude, they are rasterized, and are formed in an image data. Therefore, it can be adapted for all character fonts with small alphabetic character cache storage capacity. When such a profile alphabetic character cache has taken hierarchical storage structure with the rasterization alphabetic character cache, a still more efficient alphabetic character cache control system is realized. In this invention, further, RIP processing and font management are separated constitutionally and the interface between both can take the common format independent of a font.

[Translation done.]

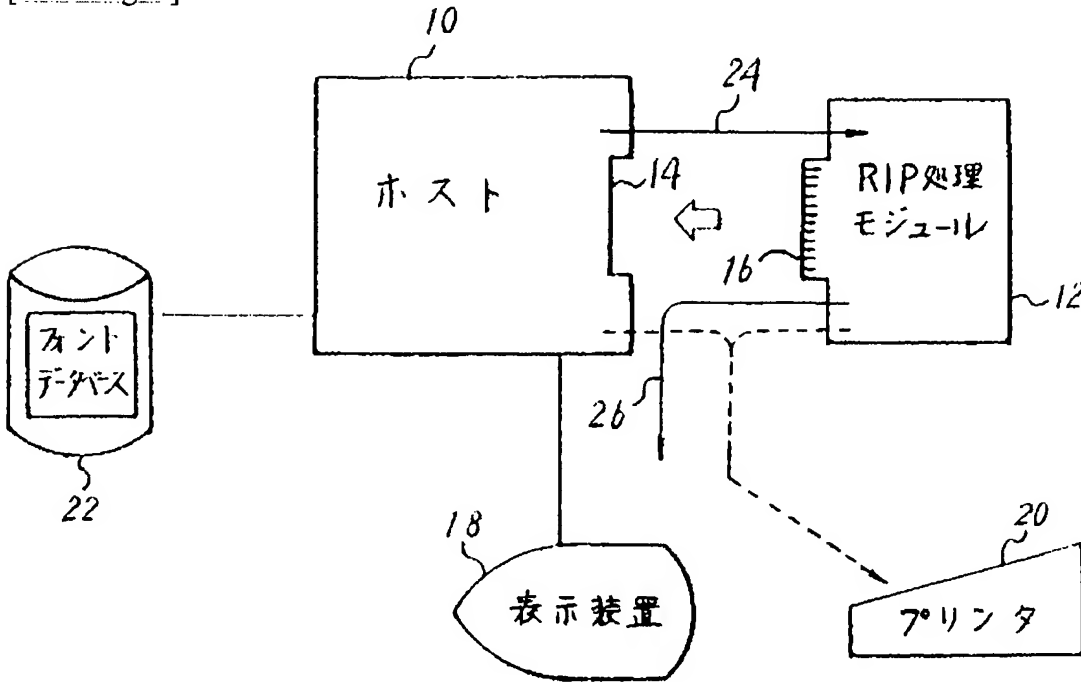
* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

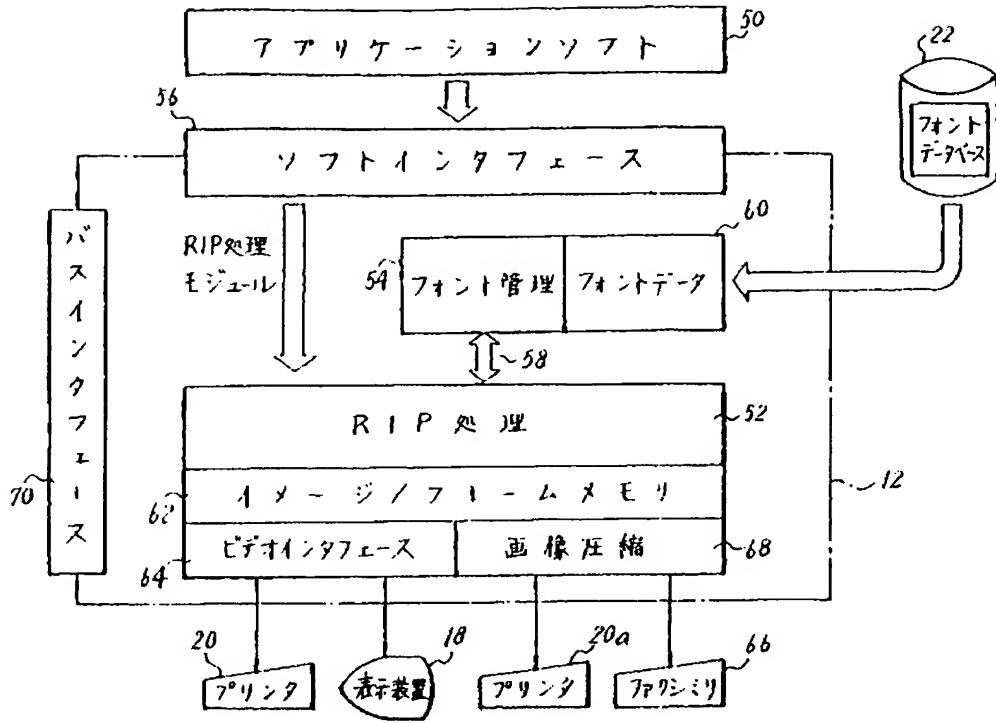
DRAWINGS

[Drawing 1]



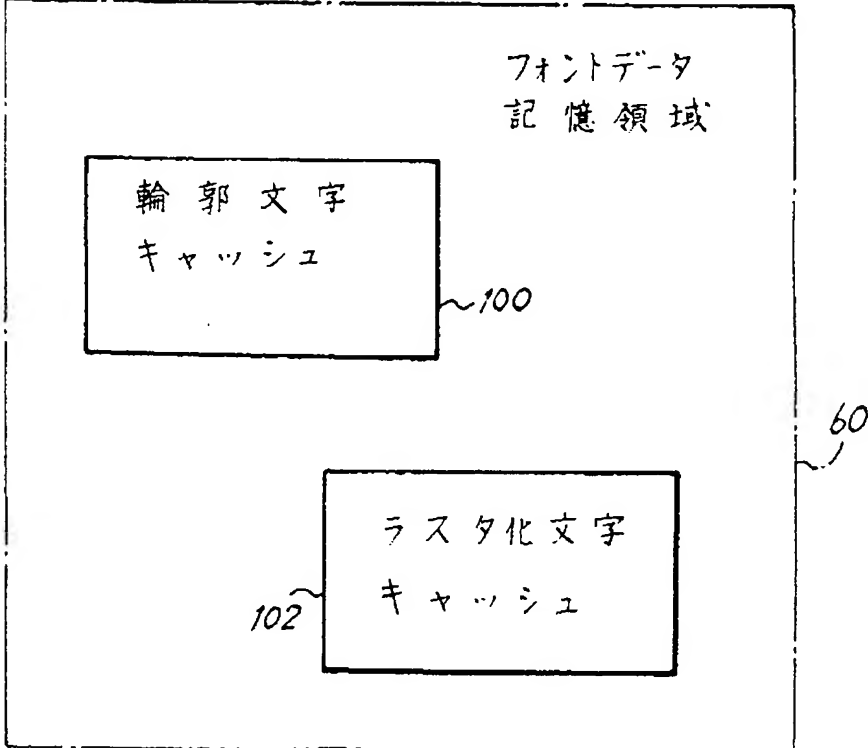
フォントフリーなラスターイメージ処理システム

[Drawing 2]



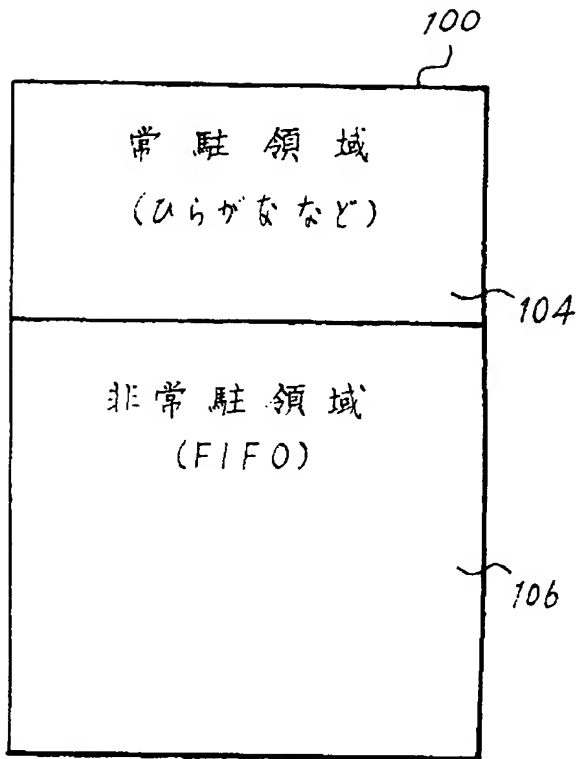
RIP処理モジュールの構成例

[Drawing 3]



文字キャッシュの構成例

[Drawing 4]



輪郭文字キャッシュの構成例

[Translation done.]

(19)日本国特許庁 (J P)

(12) 特 許 公 報 (B 2)

(11)特許番号

第2923002号

(45)発行日 平成11年(1999) 7 月26日

(24)登録日 平成11年(1999) 4 月30日

(51)Int.Cl.⁶

G 0 9 G 5/22
5/24

識別記号

6 2 0
6 7 0

F I

G 0 9 G 5/22
5/24

6 2 0
6 7 0

請求項の数 5 (全 8 頁)

(21)出願番号 特願平2-213748
(22)出願日 平成2年(1990) 8 月14日
(65)公開番号 特開平4-96095
(43)公開日 平成4年(1992) 3 月27日
審査請求日 平成7年(1995) 2 月24日

(73)特許権者 999999999
沖電気工業株式会社
東京都港区虎ノ門1丁目7番12号
(72)発明者 大宅 伊久雄
東京都港区虎ノ門1丁目7番12号 沖電
気工業株式会社内
(72)発明者 守谷 信行
東京都港区虎ノ門1丁目7番12号 沖電
気工業株式会社内
(72)発明者 畠中 啓
東京都港区虎ノ門1丁目7番12号 沖電
気工業株式会社内
(74)代理人 弁理士 香取 孝雄
審査官 大野 弘

最終頁に続く

(54)【発明の名称】 フォントフリーなラスタイメージ処理システム

1

(57)【特許請求の範囲】

【請求項1】 ページ記述言語で記述されたデータを処理してラスタイメージを形成するラスタイメージ処理システムにおいて、該システムは、前記データを受け、該データに含まれる文字のフォントデータを得るためのフォントデータ指定情報を作成し、得られたフォントデータからラスタイメージを形成するラスタイメージ処理手段と、文字フォントを表す輪郭データが各装置共通に使用できる形式で蓄積されているデータベースにフォントデータ指定情報に基づいてアクセスして、該フォントデータ指定情報に関連する輪郭データを取り込むフォント管理手段と、前記ラスタイメージ処理手段およびフォント管理手段の間をフォントに依存しない共通の形式でインタフェース

2

するインタフェース手段とを含み、前記フォント管理手段は、前記輪郭データを一時記憶する第1の蓄積手段を有し、該フォント管理手段は、前記フォントデータ指定情報に応じて、まず第1の蓄積手段にアクセスし、該フォントデータ指定情報に関連する輪郭データが第1の蓄積手段に存在すれば、これに基づいてフォントデータを生成し、該フォントデータ指定情報に関連する輪郭データが第1の蓄積手段に存在しないときは、前記データベースに該フォントデータ指定情報に基づいてアクセスして、該フォントデータ指定情報に関連する輪郭データを取り込んでこれに基づいてフォントデータを生成するとともに、該取り込んだ輪郭データを第1の蓄積手段に蓄積することを特徴とするフォントフリーなラスタイメージ処理シ

10

システム。

【請求項2】請求項1に記載のシステムにおいて、前記フォント管理手段は、

前記形成されたラスタイメージに含まれるラスタ化された文字データを一時記憶する第2の蓄積手段を有し、前記ラスタイメージ処理手段がラスタイメージを形成する際、ラスタ化された文字データを第2の蓄積手段に蓄積し、

前記フォントデータ指定情報に応じて、第1の蓄積手段へのアクセスに先立って第2の蓄積手段にアクセスし、該フォントデータ指定情報に関連するラスタ化された文字データが第2の蓄積手段に存在すれば、これに基づいてフォントデータを生成し、

該フォントデータ指定情報に関連するラスタ化された文字データが第2の蓄積手段に存在しないときは、該フォントデータ指定情報に基づいて第1の蓄積手段にアクセスすることを特徴とするラスタイメージ処理システム。

【請求項3】請求項1に記載のシステムにおいて、第1の蓄積手段は、前記輪郭データのうち予め定められた相対的に使用頻度の高いものを常駐的に蓄積する第1の記憶領域と、相対的に使用頻度の低いものを非常駐的に蓄積する第2の記憶領域とを含むことを特徴とするラスタイメージ処理システム。

【請求項4】請求項1に記載のシステムにおいて、該システムは、前記輪郭データがフォントに依存しない共通の形式で蓄積され前記フォントデータ指定情報を受けるとそれに関連する輪郭データを出力するデータベース手段を含むことを特徴とするラスタイメージ処理システム。

【請求項5】請求項1に記載のシステムにおいて、該システムは、前記ページ記述言語で記述されたデータを作成して前記ラスタイメージ処理手段に供給するホストを含むことを特徴とするラスタイメージ処理システム。

【発明の詳細な説明】

(産業上の利用分野)

本発明はラスタイメージ処理システム、より具体的には、ページ記述言語で記述されたデータを処理してラスタイメージを形成するラスタイメージ処理システムに関するものである。

(従来の技術)

文字および画像データを含むコマンドデータは、たとえばバブリッシング用アプリケーションプログラムによって作成され、ページ記述言語で記述される中間ストリームに変換される。この中間ストリームは、ラスタイメージ処理(RIP)方式によるインタプリタによってラスタイメージに変換され、プリンタや表示装置などのイメージ出力装置にページの形で出力される。

文字を扱うシステムでは、様々な言語や国に応じて多様なフォントが存在し、したがってフォントに対しては多様な要求がある。たとえば、字母、書体、大きさのみ

ならず、横書き、縦書き、左書き、右書き、斜字体、太字などの多様性がある。また、文字列すなわち文書としてフォントを扱う場合は、文字列の配列の仕方、たとえばプロポーショナル印字などの配列方式にも様々な要求がある。さらに、文字を特定するコードについては、フォントの供給者によって様々な文字コード系列が使用されている。

ラスタイメージ処理方式は、これらの多様な要求条件を少なくとも部分的に満足させねばならない。なお、本明細書において用語「フォント」は、単に字母や字体のみならず、上述のような様々な要求条件をも含む広義に解釈するものとする。

字母データは、フォント供給者に固有の形式、たとえばビットマップ形式やアウトライン(輪郭)形式などで提供される。フォントデータは一般に、文字の輪郭の直線および曲線を規定するデジタル化されたデータの形で外部ファイルやROMなどのメモリに蓄積される。ホストから特定の文字コードでフォントが指定されると、それに対応するフォントデータがファイルから読み出され、拡大または縮小などのスケーリング処理を施され、最終的にはラスタ化されてイメージに変換される。

ビットマップ形式やアウトライン形式による原データをラスタイメージに展開する処理は、フォントの形式によって相違することが多い。また、原データをスケーリングした後、長さや丸みを補正して適切な字形にするヒンティング処理も、フォントフォーマットによって異なることがある。

このようなバブリッシングの環境の下で従来は、フォントの種別に固有のラスタイメージ処理方式、すなわちフォント指向型の処理方式がとられていた。より詳細には、たとえば従来のプリンタは、ラスタイメージ処理系とフォントデータファイルを含み、セントロニクスやRS-232Cなどの汎用インタフェースでパソコンやワークステーションなどのホストに接続される。つまり、プリンタ内にあるラスタイメージ処理系とフォントデータは、特定のプリンタに固有であり、ホストはそれらを汎用のプリンタインタフェースを介して使用する構成がとられていた。また、CRTなどの画像表示装置は、ホストに接続されるが、表示装置で使用されるフォントのデータファイルと、それを扱うラスタイメージ処理系はホストの内部に含まれ、したがってホストに依存する構成がとられていた。

(発明が解決しようとする課題)

これからわかるように従来は、前述した様々な要求条件に適応的に対応できる単一のラスタイメージ処理システムが利用できなかった。より詳細には、特定のアプリケーションソフトが用いるコマンドデータには、それに固有のラスタイメージ処理系が必要であった。たとえば、PostScript(商標)などのページ記述言語によるコマンドデータストリームには、PostScriptインタプリタ

がラスタイメージ処理系として用いられる。つまり、特定のラスタイメージ処理系には、特定のフォント形式、フォント処理およびフォント読出し機能が組み込まれていた。

一般に、バブリッシングなどのドキュメント処理システムでは、複数のコマンドデータストリームが用いられる。したがって従来方式において、複数のデータストリームの扱いを可能にするためには、それぞれのコマンドストリームごとにフォントを用意しなければならなかった。これは、フォントの処理および管理システムの重複

作成を招き、フォントの重複投資を誘起していた。ところで、文字の表示を高速化するために従来は、一度使用されラスタイメージ化された文字データをキャッシュに格納し、以降、同じ文字データを表示出力する際には、キャッシュからこれを読み出して使用するシステムがあった。これによって、複数回使用される文字データは、フォントデータをラスタイメージ化する処理を行わなくて済む。

しかしこの従来方式では、キャッシュに蓄積されるラスタイメージ化文字データは、サイズがその使用された状態に固定される。したがって、次に同じ文字フォントを異なったサイズで使用するときは、キャッシュに蓄積されている文字データを使用できず、別途、文字フォントデータをファイルよりフェッチしなければならなかった。仮りに、同じ文字フォントについて複数のサイズでラスタイメージ化された文字データをキャッシュに蓄積するようにシステムを構成したとすると、当然ながら多くの記憶領域を必要とするであろう。この傾向は、漢字のように字種の多いフォントについては著しい。

本発明はこのような従来技術の欠点を解消し、高速な文字表示を行なうことができ、フォントに依存しないラスタイメージ処理システムを提供することを目的とする。

(課題を解決するための手段)

本発明によれば、ページ記述言語で記述されたデータを処理してラスタイメージを形成するラスタイメージ処理システムは、前記データを受け、このデータに含まれる文字のフォントデータを得るためのフォントデータ指定情報を作成し、得られたフォントデータからラスタイメージを形成するラスタイメージ処理手段と、文字フォントを表わす輪郭データが各装置共通に使用できる形式で蓄積されているデータベースにフォントデータ指定情報に基づいてアクセスして、フォントデータ指定情報に関連する輪郭データを取り込むフォント管理手段と、ラスタイメージ処理手段およびフォント管理手段の間をフォントに依存しない共通の形式でインタフェースするインタフェース手段とを含み、フォント管理手段は、輪郭データを一時記憶する第1の蓄積手段を有し、フォント管理手段は、フォントデータ指定情報に応じて、まず第1の蓄積手段にアクセスし、フォントデータ指定情報に関連する輪郭データが第1の蓄積手段に存在すれば、こ

れに基づいてフォントデータを生成し、フォントデータ指定情報に関連する輪郭データが第1の蓄積手段に存在しないときは、データベースにフォントデータ指定情報に基づいてアクセスして、フォントデータ指定情報に関連する輪郭データを取り込んでこれに基づいてフォントデータを生成するとともに、取り込んだ輪郭データを第1の蓄積手段に蓄積する。

また本発明によれば、フォント管理手段は、形成されたラスタイメージに含まれるラスタイメージ化された文字データを一時記憶する第2の蓄積手段を有し、ラスタイメージ処理手段がラスタイメージを形成する際、ラスタイメージ化された文字データを第2の蓄積手段に蓄積し、フォントデータ指定情報に応じて、第1の蓄積手段へのアクセスに先立って第2の蓄積手段にアクセスし、フォントデータ指定情報に関連するラスタイメージ化された文字データが第2の蓄積手段に存在すれば、これに基づいてフォントデータを生成し、フォントデータ指定情報に関連するラスタイメージ化された文字データが第2の蓄積手段に存在しないときは、フォントデータ指定情報に基づいて第1の蓄積手段にアクセスする。

さらに本発明によれば、第1の蓄積手段は、輪郭データのうち予め定められた相対的に使用頻度の高いものを常駐的に蓄積する第1の記憶領域と、相対的に使用頻度の低いものを非常駐的に蓄積する第2の記憶領域とを含む。

(作用)

本発明によれば、ページ記述言語で記述されたデータがホストからラスタイメージ処理システムに入力されると、ラスタイメージ処理手段は、このデータに含まれる文字のフォントデータを得るためのフォントデータ指定情報を作成する。フォント管理手段は、フォントデータ指定情報に応じて、まず第1の蓄積手段にアクセスする。この輪郭データが第1の蓄積手段に存在すれば、これに基づいてフォントデータを生成する。しかし、存在しないときは、データベースにフォントデータ指定情報に基づいてアクセスして、そのフォントデータ指定情報に関連する輪郭データを取り込み、これに基づいてフォントデータを生成する。これとともに、取り込んだ輪郭データを第1の蓄積手段に蓄積する。

ラスタイメージ処理手段は、こうして得られた輪郭データからラスタイメージを形成する。形成されたラスタイメージは、最終的にはラスタイメージ出力装置や通信装置へ出力される。

(実施例)

次に添付図面を参照して本発明によるラスタイメージ処理システムの実施例を詳細に説明する。第1図を参照すると、この実施例は基本的には、たとえばパソコンやワークステーションなどの処理システムからなるホスト10と、ラスタイメージ処理(RIP)を行なうモジュールパッケージすなわちRIP処理モジュール12を含み、両者

がコネクタ14および16で相互に接続される。ホスト10は、文字および画像データを含むコマンドデータを、たとえばバブリッシング用アプリケーションプログラム50（第2図）によって作成し、ページ記述言語で記述される中間ストリームに変換する、本実施例では汎用の処理装置である。

ホスト10には、たとえばCRTなどの文字および画像を可視表示する画像表示装置18が接続されている。またホスト10および（または）RIP処理モジュール12にはプリンタ20が接続されている。プリンタ20は、文字および画像を記録媒体に可視記録する画像出力装置である。勿論、ホスト10は通常の処理システムのようにキーボードなどの入出力装置も含むが、これらは本発明の理解に直接関係ないので図示と説明を省略する。

本システムで特徴的なのは、表示装置18およびプリンタ20で文字の可視化に共通に使用されるフォントデータが、たとえば固定ディスク、フロッピーまたは集積回路ボードなどの外部ファイル22に格納されていることである。つまり、これらのフォントデータは、表示装置18やプリンタ20に依存せず、データベースとして共通に利用できる。勿論、外部ファイル22は、ホスト10に固有の記憶装置のみならず、たとえば通信回線を通して遠方のデータベースにアクセスできるものでもよい。したがって、一方ではフォントがRIP処理系に依存せず、たとえばローマンから漢字までの広範囲な拡張性があるとともに、他方ではRIP処理系が文字コード体系から独立される。

RIP処理モジュール12は、後に詳述するように、RIP処理システムを含む。この点も本実施例の重要な特徴のひとつである。ホスト10は、様々なバブリッシング用アプリケーションソフト50を実行することができる。したがってRIP処理モジュール12には、それらのアプリケーションソフト50に応じて様々な種類のコマンドデータストリーム24が入力される。また、外部ファイル22のフォントデータは、RIP処理モジュール12での必要に応じてダイナミックに読み出され、RIP処理モジュール12にロードされる。RIP処理モジュール12は、コマンドデータストリーム24にその種類に応じた適切なラスタイメージ処理を施し、ラスタイメージデータ26としてホスト10またはプリンタ20へ出力する。このデータ26は、データ圧縮された形であっても、または圧縮されない形であってもよい。

RIP処理モジュール12は、その特定の構成を第2図に示すように、基本的にはRIP処理52とフォント管理54の2つの機能部を含む。アプリケーションソフト50はソフトインタフェース56でRIP処理52とインタフェースされる。

RIP処理52は、特定のアプリケーションソフト50で指定された文字のフォントデータを取り出すのに必要なフォントデータ指定情報を作成する。このフォントデータ

指定情報には、本実施例では、フォント供給者の指定に基づく書体、文字コード体系、文字コード、文字サイズなどが含まれる。書体では、たとえば漢字であれば明朝体やゴシックなどの別、文字コード体系では、日本語であればJIS83やシフトJISの別などが指定される。

RIP処理52とフォント管理54との間のインタフェース58は、フォントに依存しない共通化されたインタフェースをとる。インタフェース58は、上述のフォントデータ指定情報、およびフォントデータを含む。フォントデータは、要求に応じて外部ファイル22のデータベースからモジュール12内の領域60にロードされる。したがって、フォント管理54におけるフォントアクセスパスは、文字コード体系に応じて最適なパスをとることができる。フォント管理54は、フォントデータ60をそのままフォントデータの形で、または文字サイズデータで要求されるサイズまで拡大もしくは縮小するなどのフォント処理を行なって、フォントデータをRIP処理52へ出力する。

フォントデータの記憶領域60は本実施例では、第3図に示すように、輪郭文字キャッシュ100およびラスタ化文字キャッシュ102の2つの高速記憶領域を含む。輪郭文字キャッシュ100は、フォントデータベース22から読み込まれた文字フォントの輪郭を表わすデータすなわち輪郭データが一時蓄積される領域である。この輪郭データは、基本的にはサイズフリーであり、後に所望の大きさのラスタ化された文字データに変換されて使用される。

ラスタ化文字キャッシュ102は、輪郭文字キャッシュ100に格納された輪郭データが所望の大きさのイメージにラスタ化された結果の文字データが蓄積される記憶領域である。ラスタ化文字キャッシュ102に格納されるイメージデータは、所望の大きさに変倍されたものである。なおラスタ化は、後述のようにRIP処理52で行なわれる。このように本実施例では、フォントデータベース22から読み込まれた輪郭データは輪郭文字キャッシュ100に格納され、次にラスタ化された文字データはラスタ化文字キャッシュ102に格納されるように、文字キャッシュが2つのレベルに階層化されている。しかし、ラスタ化文字キャッシュ102は必ずしも設けられなくてもよく、文字キャッシュについては輪郭文字キャッシュ100のみを含むシステム構成をとってもよい。

輪郭文字キャッシュ100はさらに、第4図に示すように、本実施例では2つの領域、すなわち常駐領域104および非常駐領域106を含む。常駐領域104は、たとえば日本語システムではひらがなのように相対的に使用頻度の高い文字フォントの輪郭データを蓄積する記憶領域である。使用頻度の高い文字の輪郭データは、たとえばシステムの初期設定の際、フォントデータベース22から常駐領域104に読み込まれる。常駐領域104に読み込まれた輪郭データは、常にこの領域104に保持され、使用される。

非常駐領域106は、相対的に使用頻度の低い文字フォントの輪郭データを蓄積する記憶領域である。たとえば漢字の文字データは、特定の漢字が初めてフォントデータベース22にアクセスされたときにこの領域106に蓄積される。非常駐領域106の記憶容量は有限である。そこで、たとえば最も最近に読み込まれたものから順に蓄積し、記憶容量いっぱい蓄積されているときは最も古く読み込まれた文字データを消去するFIFO動作で、文字単位に読書き制御を行なう方法がある。または、非常駐領域106に格納されている輪郭データのそれぞれの使用回数を計数し、使用回数の最も少ないデータから順に消去するように構成してもよい。

フォント管理54は本実施例では、1文字単位の管理を基本とするのが望ましい。これは、従来技術では個々のアプリケーションソフトに応じたコマンドストリームで有していたフォントデータのアクセス機構を、本実施例ではフォント管理54で吸収させるためである。フォントデータをスケールリングすると、文字の輪郭を形成している線分の長さが不揃いになったり、丸みに滑らかさが欠けたりすることがある。そのような場合、長さや丸みを補正して適切な字形にするヒンティング処理もフォント管理54にて行なわれる。

RIP処理52は、フォント管理54から受けたフォントデータをラスタ化する。ラスタ化されたイメージデータは、イメージ/フレームメモリ62に蓄積され、ページ単位の画像データに組み立てられる。イメージ/フレームメモリ62は、たとえばページ単位でラスタイメージデータを一時蓄積する記憶領域である。

イメージ/フレームメモリ62は、ビデオインタフェース64を介してプリンタ20および表示装置18とインタフェースされる。また、圧縮データ形式でインタフェースされるプリンタ20aやファクシミリ66などの出力装置に対しては、画像圧縮68によってインタフェースされる。ホスト10に対してRIP処理モジュール12はバスで接続され、このバス接続はハードウェアバスインタフェース70を介して行なわれる。

動作を説明する。まず、装置に電源を投入すると、RIP処理52はフォントデータベース22にアクセスして、相対的に高い使用頻度の文字フォント、たとえば仮名の輪郭文字データを輪郭文字キャッシュ100に読み込み、これらをその常駐領域104に蓄積し、後の使用に備える。

文字および画像データを含むコマンドデータは、アプリケーションソフト50によって作成され、ページ記述言語で記述される中間ストリームに変換される。この中間ストリームは、ソフトウェアインタフェース56を介してRIP処理52に入力される。RIP処理52では、このコマンドストリームに含まれる文字のフォントデータにアクセスするのに必要なフォントデータ指定情報を作成し、これをフォント管理54に与える。

フォント管理54は、フォントデータ指定情報に基づい

て、まず、ラスタ化文字キャッシュ102にアクセスする。必要な文字サイズのラスタ化文字データがキャッシュ102に存在すれば、フォント管理54はそれを取り込む。必要な文字サイズのラスタ化文字データがキャッシュ102になければ、フォント管理54は輪郭文字キャッシュ100にアクセスし、そのデータがキャッシュ100に存在すれば、それを取り込む。たとえばひらがなは、本実施例では常にこのキャッシュ102から輪郭データの形で入手される。必要な輪郭データが両キャッシュ100および102のいずれにも存在しないときは、フォント管理54は、外部ファイル22のフォントデータベースにアクセスし、必要なフォントデータを取り込むとともに、輪郭文字キャッシュ100にも蓄積して後の利用に備える。たとえば漢字など、比較的使用頻度の低い文字フォントについては、その輪郭データは輪郭文字キャッシュ100の非常駐領域106に読み込まれる。

フォント管理54は、輪郭文字キャッシュ100のフォントデータをそのままフォントデータの形で、または必要なサイズまで拡大もしくは縮小し、RIP処理52へ出力する。仮名などの使用頻度の高い輪郭データは常駐領域104から、その他の輪郭データは非常駐領域106から読み出される。

RIP処理52では、フォント管理54から受けたフォントデータをラスタ化し、文書データとして一旦、イメージ/フレームメモリ62に蓄積する。その際、ラスタ化された文字フォントは、フォントデータ記憶領域60のラスタ化文字キャッシュ102にも格納し、以降の使用に備える。こうしてイメージ/フレームメモリ62には、ページ単位の画像データが組み立てられる。

イメージ/フレームメモリ62に蓄積されたラスタイメージは、プリンタ20または表示装置18に出力するときには、ビデオインタフェース64を介して出力され、プリンタ20a、またはファクシミリ66などの通信装置に出力するときは、画像圧縮68によって符号化圧縮される。こうしてアプリケーションソフト50からのコマンドデータストリームはRIP処理モジュール12によってラスタイメージに変換され、たとえばプリンタ20や表示装置18などのイメージ出力装置にページの形で出力される。

本実施例ではこのように、輪郭文字キャッシュ100が設けられ、フォントデータはラスタ化される前の輪郭文字データの形でこれに蓄積される。この輪郭文字データは、後に使用の必要の都度アクセスされ、所望の大きさに変倍されてラスタ化され、イメージデータとして出力される。したがって、従来のラスタ化キャッシュのみが設けられた方式に比較して、少ない文字キャッシュ記憶容量で、あらゆる文字サイズに適應できる。本実施例ではまた、ラスタ化文字キャッシュ102とともに輪郭文字キャッシュ100が階層構造をとっているため、効率のよい文字キャッシュが実現される。

(発明の効果)

11

本発明によればこのように、文字フォントの輪郭データを一時記憶するキャッシュが設けられ、フォントデータはラスタ化される前の形でこれに蓄積される。キャッシュに蓄積された輪郭データは、後に使用の必要の都度アクセスされ、所望の大きさに変倍されてラスタ化され、イメージデータに形成される。したがって、少ない文字キャッシュ記憶容量で、あらゆる文字フォントに対応できる。このような輪郭文字キャッシュがラスタ化文字キャッシュとともに階層メモリ構造をとっている場合は、さらに効率のよい文字キャッシュ制御方式が実現される。本発明ではさらに、RIP処理とフォント管理が構成上、分離され、両者の間のインタフェースはフォントに依存しない共通の形式をとることができる。

【図面の簡単な説明】

第1図は本発明によるフォントフリーなラスタイメージ処理方式の実施例の全体構成を示す機能ブロック図、第2図は、第1図に示す実施例におけるRIP処理モジュールの構成例を示す機能ブロック図、

*

12

* 第3図は同実施例における文字キャッシュの構成例を示す概念図、

第4図は同実施例における輪郭文字キャッシュの構成例を示す概念図である。

主要部分の符号の説明

10……ホスト

12……RIP処理モジュール

18……表示装置

22……フォントデータベース

10 52……RIP処理

54……フォント管理

60……フォントデータ記憶領域

62……イメージ/フレームメモリ

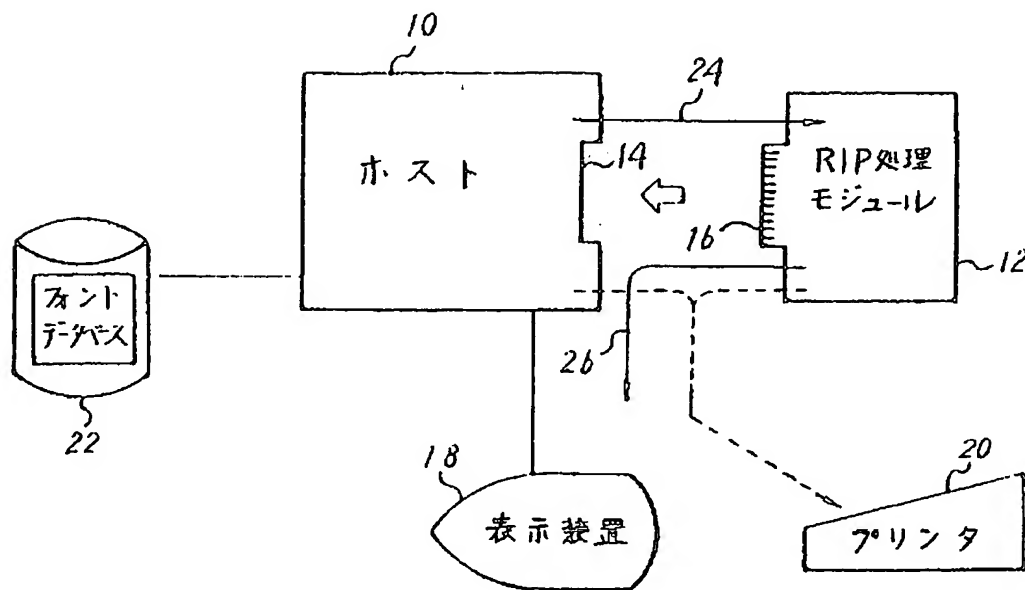
100……輪郭文字キャッシュ

102……ラスタ化文字キャッシュ

104……常駐領域

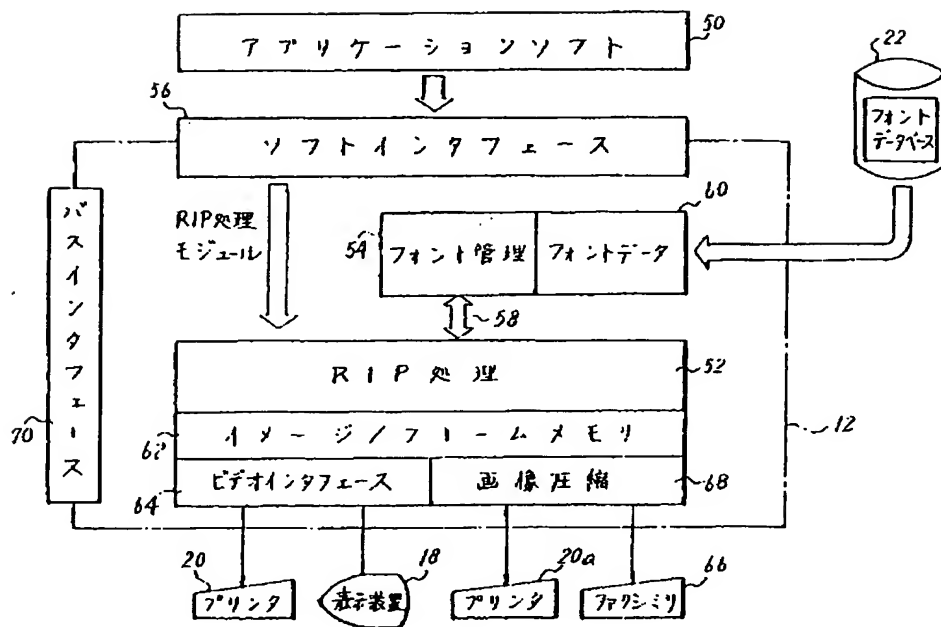
106……非常駐領域

【第1図】



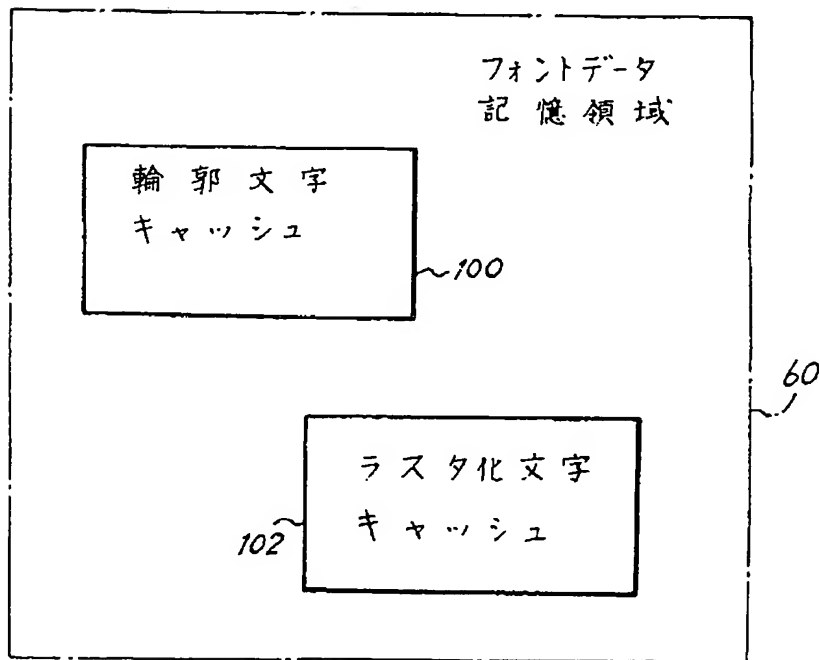
フォントフリーなラスタイメージ処理システム

【第2図】



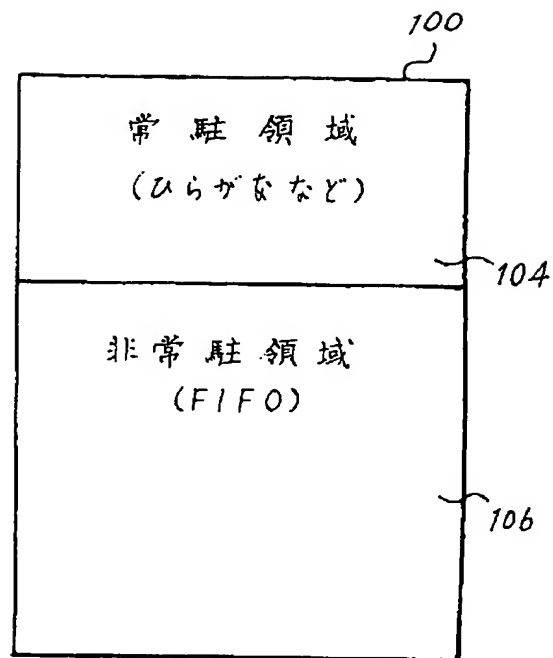
RIP処理モジュールの構成例

【第3図】



文字キャッシュの構成例

【第4図】



輪郭文字キャッシュの構成例

フロントページの続き

(72)発明者 小山 法孝
東京都港区虎ノ門1丁目7番12号 沖電
気工業株式会社内

(56)参考文献 特開 平1-275057(JP, A)
特開 平1-188354(JP, A)

(58)調査した分野(Int.Cl.⁶, DB名)
G09G 5/22

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-096095

(43)Date of publication of application : 27.03.1992

(51)Int.Cl.

G09G 5/22

G06F 15/62

G09G 5/40

(21)Application number : 02-213748

(71)Applicant : OKI ELECTRIC IND CO LTD

(22)Date of filing : 14.08.1990

(72)Inventor : OYA IKUO

MORIYA NOBUYUKI

HATANAKA HIROSHI

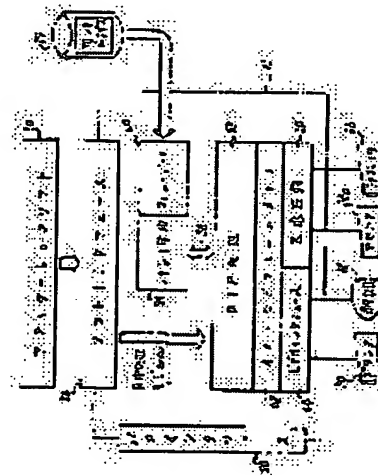
KOYAMA NORITAKA

(54) FONT-FREE RASTER IMAGE PROCESSING SYSTEM

(57)Abstract:

PURPOSE: To display characters fast by generating font data specification information according to data which are described in page description language, and referring to a cache or data base and generating a font contour before raster image processing.

CONSTITUTION: When the data which are described in the page description language are inputted from a host, a raster image(RIP) module 12 executes application software 50 to generate the font data specification information for obtaining font data on characters in the data. A font control means 54 accesses the font data storage area 60 of the cache according to the font data specification information to generate font data according to contour data. When no contour is present, the data base 22 is accessed to fetch contour data and font data is generated according to the contour data. At the same time, the fetched contour data are stored in a 1st storage means. The font contour is generated before the raster image processing to display and print characters at a high speed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C): 1998,2003 Japan Patent Office